

# On the Distributed Construction of a Collision-Free Schedule in Multi-Hop Packet Radio Networks

Jaume Barcelo · Boris Bellalta · Cristina Cano · Azadeh Faridi · Miquel Oliver

Received: date / Accepted: date

**Abstract** This paper introduces a protocol that distributively constructs a collision-free schedule for multi-hop packet radio networks in the presence of hidden terminals. As a preliminary step, each wireless station computes the schedule length after gathering information about the number of flows in its neighbourhood. Then, a combination of deterministic and random backoffs are used to reach a collision-free schedule. A deterministic backoff is used after successful transmissions and a random backoff is used otherwise. It is explained that the short acknowledgement control packets can easily result in channel time fragmentation and, to avoid this, the use of link layer delayed acknowledgements is advocated and implemented. The performance results show that a collision-free protocol easily outperforms a collision-prone protocol such as Aloha. The time that is required for the network to converge to a collision-free schedule is assessed by means of simulation.

**Keywords** MAC protocol · collision-free schedule · multi-hop packet radio networks

## 1 Introduction

Most of the wireless networks in use nowadays rely on some kind of infrastructure. A clear example is IEEE 802.11 networks in which wireless stations connect to

wireless access points and access points are used as gateways to the wired network. These networks use wireless communications only for the last hop and present some limitations. To start with, it is difficult to deploy a network in those places where there is no infrastructure available. It is envisioned that, in the near future, multi-hop packet radio networks should offer networking capabilities in places where network infrastructure is not available.

A packet radio network is defined in [30] as a collection of wireless stations that exchange messages via broadcast radio. Each wireless station (often referred to as wireless node, or simply station or node) consists of a radio and a controller. The controller is capable of executing routing algorithms and therefore the stations can forward packets that belong to other stations.

Multi-hop packet radio networks present several challenges and interdependencies at all layers of the protocol stack. Our goal in the present work is to focus on the media access control (MAC) layer and explore the possibility of distributively constructing a collision-free schedule in multi-hop packet radio networks. A collision is simply the interference caused when there are two active transmitters in the neighborhood of a receiver. Collisions can be avoided by a careful scheduling of all the transmissions in the network, and the construction of such schedule in a distributed fashion is the object of the present work.

This paper is the extended version of a workshop paper where the main idea was originally outlined [7]. This idea can be summarized as using a deterministic backoff after successful transmissions and a random backoff otherwise. Each terminal uses random backoffs until it finds a free transmission opportunity where no other terminal is transmitting over the shared channel. It will then periodically transmit during that same transmis-

---

J. Barcelo  
Universidad Carlos III de Madrid  
E-mail: jaume.barcelo@uc3m.es

B. Bellalta, C.Cano, A. Faridi, M. Oliver  
Universitat Pompeu Fabra  
E-mail: {boris.bellalta; cristina.cano; azadeh.faridi; miquel.oliver}@upf.edu

sion opportunity with a period equal to the schedule length. If there is a collision, the nodes involved perform random backoffs until they each find another available transmission opportunity. Once all nodes find their transmission opportunities within the schedule length, there will be no further collisions.

Compared to the original workshop paper, the present work offers the following additional contributions:

- An extended and updated literature review, which includes the latest advancements in this area.
- A modification of the protocol to support link layer delayed acknowledgements (ACKs).
- New simulation results that show that the modified protocol offers a shorter transient state.

As mentioned before, the fundamental technique we rely on is the use of a deterministic backoff after successful transmissions and a random backoff otherwise. This idea has already been used in other problems, as it is described in Sect. 2, which reviews previous work in this area and offers the necessary background for the remainder of the paper. The problem of channel time fragmentation that results from the use of link-layer ACKs is introduced in Sect. 3. This problem can be alleviated if delayed ACKs are used at the link layer. Then, in Sect. 4 we cover the protocol that the different stations follow to distributively compute the schedule length and then cooperatively construct a collision-free schedule for wireless transmissions. Performance results are presented in Sect. 5, where the aggregate throughput and the duration of the transient state is evaluated for different schedule lengths using an example topology. Further refinements and aspects that are considered out of the scope of this article are discussed in Sect. 6. Finally, Sect. 7 concludes the paper.

## 2 Related work

The problems associated to multi-hop packet radio networks (also known as multi-hop radio networks or wireless mesh networks in recent literature) have attracted the attention of the research community for a long time. As early as 1987, Tobagi presented a survey on the modeling and analysis of multi-hop packet radio networks [30], where he identified the difficulties intrinsic to this kind of networks. First, because of the broadcast and shared nature of the wireless channel, medium arbitration protocols are needed. Moreover, the action of each wireless station inevitably affects several surrounding stations, thus incurring interdependencies that complicate the analysis.

To further complicate the issue, wireless propagation introduces an additional degree of randomness and

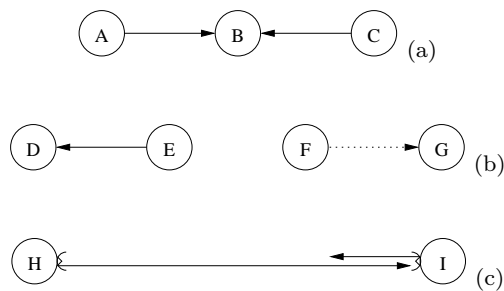
unpredictability to the network. In order to advance in the analysis of multi-hop packet radio networks, some simplifications are in order. One of the most common modeling assumptions is the use of a graph representation of the network, where an edge exists between two stations if they are in the transmission range of each other. As an example, [2] uses a graph representation to show that determining whether a traffic matrix belongs to the capacity region of a packet radio network is NP hard. We will use the graph topology representation and several other assumptions in the present paper.

Our goal is that the different stations of the network settle down in a satisfactory transmission schedule that efficiently uses the radio resources. In [27] the problem is treated from a theoretical perspective in which the network topology is known and the schedule can be constructed in a centralized way. An interesting alternative is considered in [31] where contention MAC protocols are proposed in which the stations learn from their neighbourhood. These two works rely on a slotted channel access, which means that neighbouring stations need to have closely synchronized clocks. The problem of gradient clock synchronization is treated in [28].

A possible solution to avert the problems of network synchronization and schedule construction is to use random media access control. If all the stations use the Aloha protocol, the contention parameter can be optimized to maximize proportional fairness as demonstrated in [20]. The contention parameter regulates how aggressively an Aloha station contends for the medium. If the contention parameter is too aggressive, a large fraction of channel time is wasted in the form of collisions. If it is not aggressive enough, a large fraction of the channel time remains idle.

Another alternative, which is the one considered in the present work, is the use of learning MAC protocols to distributively construct a collision-free schedule. This approach has already been exploited in the area of wireless local area networks [3]. Specifically, [3] suggests the use of a deterministic backoff after successful transmissions to prevent collisions. The advantage of [3] compared to previous solutions to prevent collisions (e.g., [11, 29]) is that it requires minimum changes to the protocol and it is backward compatible with existing implementations [9].

Later works have explored the validity of the idea for different kinds of traffic [8], modelled the network performance metrics [4], and studied the possibility of traffic differentiation [6]. A comprehensive simulation study, together with a model of the learning process is presented in [16]. This last paper also includes performance measures in non-ideal conditions, such as the presence of legacy stations or channel errors. More re-



**Fig. 1** Problems arising when using CSMA in multi-hop packet radio networks and long-distance wireless networks. (a) the hidden terminal effect. (b) the exposed terminal effect. (c) the distant terminal effect.

sults regarding the learning process and fairness with legacy stations can be found in [9]. An evaluation of a collision-free protocol and its interaction with the autorate fallback mechanism (ARF) in realistic multiple-input-multiple-output (MIMO) channels is presented in [24].

A general discussion of this class of protocols, together with performance comparisons and protocol refinements, is offered in [14]. This last paper also introduces the concepts of stickiness and variable schedule length to accommodate a larger number of contenders in slotted networks. In the present work, we will reuse both concepts in the context of multi-hop packet radio networks.

Collision-free MAC protocols for multi-hop packet radio networks have already been considered in [18]. However, it was under the assumption of slotted channel time. In the present paper we study networks in which slot synchronization is not available.

The general problem of constructing a collision-free schedule belongs to the family of decentralized constraint satisfaction problems [13]. Other problems of this family include channel selection in WLANs and inter-session network coding. The same principle can be used for channel selection in cognitive networks as in [22].

From a more practical point of view, many of the multi-hop packet radio networks deployed so far (e.g., [10] and [26]) use IEEE 802.11 equipment that relies on carrier sense multiple access with collision avoidance (CSMA/CA) for media access control. However, it has been shown that CSMA/CA performs poorly in multi-hop packet radio networks because potential interferers are often beyond the carrier sense range [15].

In Fig. 1 we describe three of the problems that are associated with the use of CSMA/CA in multi-hop packet radio networks. The stations are labelled circles and transmissions are represented as arrows.

The first problem is the *hidden terminal problem* (Fig. 1(a)) which occurs when two stations that cannot carrier-sense each other simultaneously transmit. The intended recipient of the transmissions is a third station (labelled *B* in the figure) that cannot recover either one of the two transmissions, since they overlap in time.

A complementary problem is the *exposed terminal problem* which arises when two stations avoid to transmit simultaneously even though they do not interfere with each other. This is illustrated in Fig. 1(b), where station *F* wants to transmit to *G* and it is waiting for *E* to finish. In this particular scenario, *D* and *F* are not in each other's transmission range, and the same is true for *E* and *G*. Therefore, the fact that *F* is deferring its transmission implies an inefficient use of the radio channel resources.

The last problem is the *distant terminal problem* which is depicted in Fig. 1(c). The two stations *H* and *I* are separated by a very long distance. At a given time *H* senses the channel idle and starts a transmission. While the radio signal is traversing the long distance from *H* to *I*, station *I* still senses the channel idle and starts a transmission to *H*. When *H*'s signal finally arrives to *I*, station *I* is transmitting and therefore *H*'s transmission is lost due to collision.

As CSMA suffers from these three inefficiencies, the protocol that we suggest in Sec. 4 does not make use of carrier sense information to build the collision-free schedule for multi-hop packet radio networks.

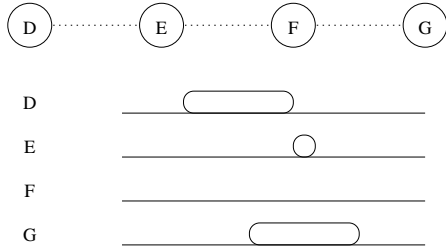
The interest on multi-hop packet radio networks has resulted in a standardization effort described in the IEEE 802.11s standard amendment [17]. Regarding the MAC layer, IEEE 802.11s offers two alternatives: the compulsory Enhanced Distributed Channel Access (EDCA) and the optional Mesh Coordinated Channel Access (MCCA).

EDCA is a CSMA mechanism that suffers from the aforementioned three inefficiencies. An assessment of the performance of an IEEE 802.11s network that supports voice-over-IP (VoIP) flows is presented in [1].

To prevent the hidden terminal problem, MCCA uses channel reservation. However, reservation is not the ultimate solution to interference. Firstly, the reservation needs to be done on the same channel that is in risk of collisions due to the hidden terminals. Therefore, it is likely that the reservation suffers a collision and consequently the data packets also collide. Secondly, in the case of successful reservation, interference can still occur because of layer 2 ACKs [23]. We will devote some discussion to ACKs in the next section and suggest the use of delayed ACKs to alleviate this problem. Thirdly, another source of interference is the presence of traffic flows beyond the two-hop neighbourhood [12]. In that

**Table 1** This table presents a comparison with previous work. Even though [17] uses slots, we mark it as non-slotted as the slots in IEEE 802.11s are of local significance only.

paper	collision-free	multi-hop	non-slotted
[3],[14]	yes	no	no
[18]	yes	yes	no
[17]	no	yes	yes
present paper	yes	yes	yes



**Fig. 2** A short ACK packet sent by *E* interferes with the packet that is being received at *F*.

paper it is suggested to change the MCCA schedule when interference is detected. Our approach also combats this kind of interference, since any unsuccessful transmission triggers a schedule change until collision-free operation is reached.

Compared to previous work (see Table 1), our proposal is the first that simultaneously satisfies the following three requirements:

- It achieves collision-free operation.
- It operates in multi-hop packet radio networks.
- It does not require a slotted channel.

This paper presents the basic core principles of collision-free operation in multi-hop packet radio networks. We describe a bare-bones protocol to illustrate the possibility of collision-free operation in networks in which the channel time is not slotted. There are many possible refinements to be made on top of this minimalist protocol and we mention some of the most promising lines of research in Sect. 6.

### 3 Link layer delayed ACK in multi-hop packet radio networks

Given the uncertainty of wireless communications, it is common in wireless networks for each link layer unicast transmission to be followed by an ACK. The problem is that, in the presence of hidden terminals, this short ACK control frame can interfere with long data packets transmitted by other terminals hidden to it, thus causing a collision and a considerable waste of channel time.

To illustrate this problem we use a topology with four stations in a row as in Fig. 2, where stations that

are within hearing range of each other are connected with a dashed line. Station *D* starts a transmission intended for station *E*. Some time later, *G* starts a transmission intended for station *F*. As soon as *D*'s transmission finishes, *E* replies with a short ACK. This short control frame interferes with *G*'s transmission which is not correctly received by station *F* and therefore, it is not acknowledged.

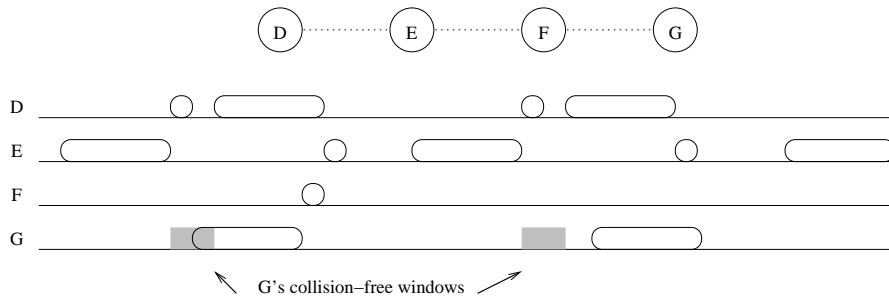
The fact that each unicast data frame needs to be acknowledged, duplicates the number of transmissions in the packet radio network. Furthermore, a packet needs to be retransmitted if either the data or acknowledgement frames are lost. This means that the mere existence of ACKs increases the chances of collisions and retransmissions and, as a consequence, reduces the network performance.

This problem can be partially alleviated by using delayed ACK techniques at the link layer. Delayed ACK is commonly used in transport control protocols (TCP) to reduce overhead. The idea is that the ACK is delayed and piggybacked to a data packet.

We suggest the use of delayed ACK at the link layer to prevent *channel time fragmentation*. By channel time fragmentation we mean that it is difficult or impossible to find the contiguous amount of channel time required for a successful transmission. Fragmentation is a common issue in computer file systems and is mentioned in the context of multi-hop packet radio networks in [12].

To illustrate the problem of channel fragmentation we propose the following example. In Fig. 3, stations *D* and *E* are sending (and acknowledging) packets to each other. For the purpose of this example, we assume fixed transmission duration. ACKs are transmitted right after the data frame is successfully received. The figure also shows *G*'s *collision-free windows*, which are the time interval at which *G* can start a transmission to *F* that will not be interfered by *E*'s transmissions or ACKs. The collision-free windows are represented as shaded areas in the figure. As an example, the figure shows two transmissions by *G*. The transmission that starts in one of the collision-free windows is successful and is acknowledged. The second transmission by *G*, which starts outside of the collision-free windows, collides with an ACK transmitted by *E* and is lost. Notice that *G*'s collision-free windows represent only a small fraction of the channel time.

Compare the situation with Fig. 4 where delayed ACKs are used. The stations do not transmit ACK frames immediately. Instead, each station waits until it transmits a data frame and prepends the ACK to the data frame. As a result, *G*'s collision-free windows more than double, which is a very positive outcome.



**Fig. 3**  $G$ 's collision-free windows are those intervals at which  $G$  can start a transmission to  $F$  without being interfered by  $E$ 's transmissions or ACKs. In the figure, collision-free windows are represented as a shaded area.

It can also be the case that a station has several ACKs to transmit when it accesses the channel. If this is the case, the station transmits all the ACKs before transmitting its own data. Transmitting the ACKs and the data consecutively has some advantages in terms of overhead, since it avoids the need for dedicated inter-frame spaces, training sequences and headers for the ACK control packets.

In this work we study a saturation scenario, in which all the stations have always a data packet ready to be transmitted. For completeness, it is also necessary to mention the case in which a station has one or more ACKs to transmit and no data packets when it completes its backoff. That station is expected to transmit the ACKs, even though it has no data packet.

In the remainder of the paper, the use of link layer delayed ACK is assumed.

#### 4 Constructing a collision-free schedule in multi-hop packet radio networks

Collision-free schedule in multi-hop packet radio networks is advantageous in the sense that no channel time is wasted in collisions and the behaviour of the network is more predictable. A possible alternative to construct such collision-free schedule is to gather all the information of the network in a centralized point, construct a collision-free schedule, and then disseminate this schedule to the network stations.

Another alternative is to distributively construct the collision-free schedule in such a way that each station takes its own decisions using information gathered from its immediate neighbourhood. This is the approach that we use in this paper, and the protocol that we propose has two clearly differentiated phases: the first one is the computation of the schedule length and the second one is the construction of the collision-free schedule.

We will be working with saturation and bounded transmission duration assumptions in the following. Sat-

uration means that, for each one-hop data flow, the transmitter has always a packet ready to be transmitted to the receiver. This assumption can be justified by the fact that it represents the maximum load that can be placed on the network. If the network is not saturated, it means that there is more bandwidth available than the one that is actually required, and therefore the network operation is satisfactory. As soon as queues build up, the saturation assumption holds.

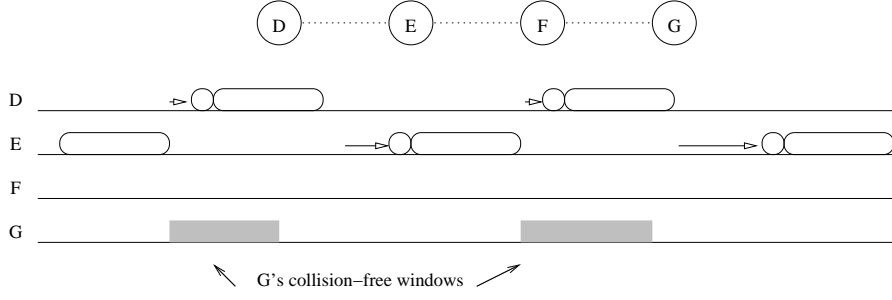
Regarding the bounded transmission duration, each station is granted a *transmission opportunity* each time it accesses the channel. A station is not allowed to occupy the channel for a time longer than the transmission opportunity. Note that a station does not have to necessarily use all of its transmission opportunity as it is just an upper bound on the time that it can access the channel.

The concept of transmission opportunity is not new and it is used in IEEE 802.11 EDCA. The existence of a transmission opportunity is natural in MAC protocols to prevent that a single station captures the channel for an extremely long time. For convenience, in this paper we will normalize the duration of a transmission opportunity to one. It should be clarified that a station has to transmit both the ACKs for the previously received packets and the data within its transmission opportunity.

##### 4.1 The schedule length

The goal of the protocol is to construct a collision-free schedule by trial and error. Using that schedule, each station knows exactly which is the right time to transmit without causing or receiving interference. Once the collision-free schedule has been constructed, it repeats in time in a periodic fashion. Each of the participating stations is given a chance to transmit at least once within the duration of the schedule.

The schedule length needs to be long enough to accommodate all incoming and outgoing flows in the



**Fig. 4**  $G$ 's collision-free windows grow when ACKs are delayed, avoiding channel fragmentation.

neighbourhood of a station. In fact, we require the schedule length to be strictly larger than the total number of flows in the neighbourhood, multiplied by the length of the transmission opportunity. This ensures that, given a randomly constructed schedule, there is a non-zero probability that this schedule is collision-free.

Note that the schedule length, as described above, does not need to be the same for all stations. Those stations that have many flows in their neighbourhood may require a longer schedule length than stations having only a few flows in their neighbourhood. Still, for the protocol to behave as intended once a collision-free schedule is constructed, it is necessary that the behaviour of the whole network is periodic with a period equal to the longest of the different schedule lengths. In the following we describe an ingenious trick that has already been used in other papers (e.g., [14], Sect. 5.3).

The goal here is to choose the schedule lengths in such a way that for any two different schedule lengths, the longer one is an integer multiple of the shorter one. To this extent, we will compute each schedule as an integer power of two multiplied by the duration of the transmission opportunity plus an arbitrarily small finite value ( $\epsilon$ ). This small value is needed to guarantee that the schedule length computed by each node is strictly larger than the time consumed by the flows in its neighbourhood.

For convenience, we normalize the transmission duration to one, and then we write

$$T_{\sigma_i} = 2^{n_i}(1 + \epsilon), \quad (1)$$

where  $T_{\sigma_i}$  is the schedule length for station  $i$ ,  $\epsilon$  is an arbitrarily small positive value, and  $n_i$  is an integer that we discuss next. We need  $2^{n_i}$  to be larger than the total number of flows in  $i$ 's neighbourhood, and thus we choose  $n_i$  as

$$n_i = \left\lceil \log_2 \left( \sum_{k \in K_i} |I_k| + |O_k| \right) \right\rceil, \quad (2)$$

where  $\lceil \cdot \rceil$  is the ceiling operator and  $K_i$  is the set of neighbours of station  $i$ .  $I_k$  and  $O_k$  are the number of incoming and outgoing flows of station  $k$ , respectively.

If we compute  $n_i$  as in (2), then  $2^{n_i}$  is the smallest power of two not smaller than the number of flows in the neighborhood of station  $i$ .

We use powers of two and logarithms of two in our approach. Any integer larger than two can be used instead of two. However, using two gives the finest granularity [14].

Note that (1) is constructed in such a way that if  $T_{\sigma_i} < T_{\sigma_j}$  for some pair of stations  $i, j$ , then  $T_{\sigma_j}$  is an integer multiple of  $T_{\sigma_i}$ , which means that the behaviour of station  $i$  is also going to be periodic with period  $T_{\sigma_j}$ . Since all the schedules evenly divide each other (just as in [14], Sect. 5.3), the behaviour of the network is periodic as a whole. The global network period is simply

$$T_\sigma = \max_i T_{\sigma_i}. \quad (3)$$

The computation of  $T_{\sigma_i}$  is very conservative in the sense that it does not take into account the fact that some flows in the same neighbourhood may overlap in time without interference. As an example, looking at Fig. 1(b), transmissions from  $E$  to  $D$  and from  $F$  to  $G$  can overlap on time. If we take station  $E$  and compute the number of flows in its neighbourhood, we observe an incoming flow in  $D$  and an outgoing flow in  $F$ . These count as two different flows even though they can overlap in time. Furthermore, a flow between two neighbours will be counted twice.

These factors can result in the computation of a schedule length that is longer than what it is strictly necessary. Even though a longer schedule translates into a lower efficiency in the steady state, it also shortens the time required to convergence to collision-free operation.

The fact that different stations use different schedule lengths introduces some difficulties when link layer delayed ACKs are used. A station that transmits to a neighbour with a larger schedule length may not receive its ACK in time, even if no collisions occur.

If two stations use the same schedule length, they will transmit alternatively one after the other and the ACKs will always arrive before the deterministic back-off expires. This is not necessarily true when the two

stations use a different schedule length, since one station may transmit more often than the other. When different schedule lengths are used, it is necessary to introduce stickiness to the backoff protocol. Stickiness is a property of learning protocols (e.g., [14]) which is discussed in subsection 4.3.

The suggested approach to compute the schedule length still requires that the stations exchange a limited amount of information with their immediate neighbours before the schedule is computed. Indeed, it requires exactly the same information as is required for the computation of the contention parameter that maximizes proportional fairness when the Aloha protocol is used [20]. It should be possible to convey this information in short beacons or *hello* messages before the schedule is computed. This information can be exchanged when the network starts to operate, and consists of very short signaling packets.

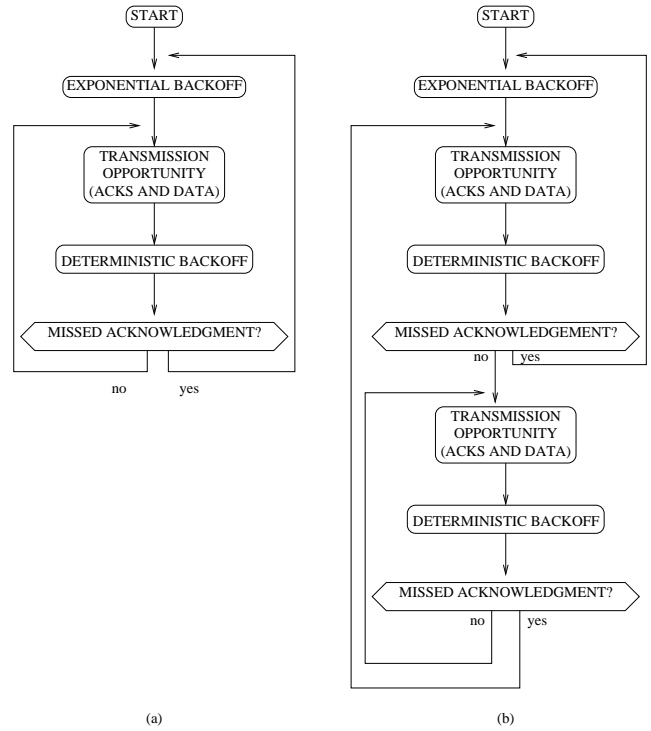
#### 4.2 The backoff protocol

After each station has computed its schedule length, it will use a very simple protocol to contend for the medium. This protocol does not use carrier sense information. It relies exclusively on the outcome of the last transmission attempt, either success or failure, to schedule the next transmission attempt.

It will defer the first transmission for a backoff time that is drawn from an exponential distribution with a parameter equal to the schedule length. When the exponential backoff expires, the station is allowed to transmit. The station is allowed to occupy the channel for the length of a transmission opportunity. Even if the station uses only a fraction of the transmission opportunity, it has to wait until the end of the transmission opportunity before proceeding with the next step of the protocol.

After the transmission opportunity ends, the station will wait for a deterministic backoff equal to the schedule length minus the duration of the transmission opportunity. When this deterministic backoff expires, the station will be granted a new transmission opportunity if the last packet has been already acknowledged. Otherwise, it will wait for an additional exponential backoff before starting the transmission opportunity.

It is important to highlight that the time that elapses from the start of a transmission opportunity to the next transmission opportunity is completely deterministic and exactly equal to the schedule length  $T_{\sigma_i}$  of the station, if both the data transmission and the ACK are successful. The operation of the protocol is summarized in the flow chart in Fig. 5(a).

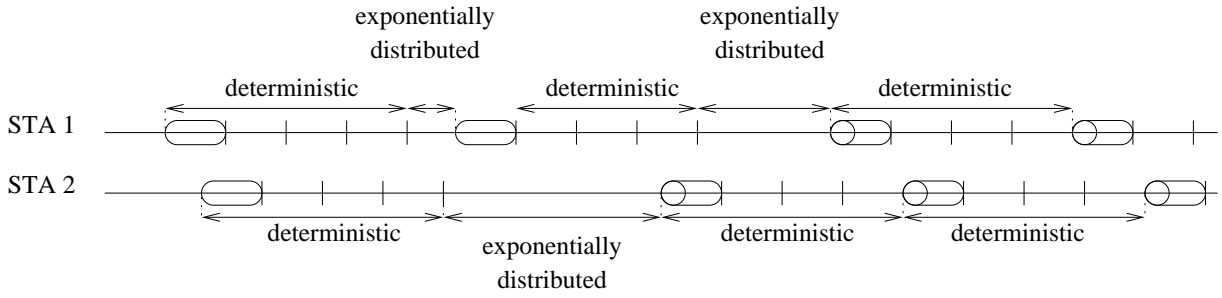


**Fig. 5** Flow chart description of the backoff protocol executed by each station. Regular version (a) and the sticky version with a degree of stickiness equal to two (b).

In each cycle, there is a transmission opportunity that each station can use to send ACKs and data. The station must go through the transmission opportunity step in all cases (even if it has nothing to transmit) and the duration of this step is deterministic. The next step is the deterministic backoff with a duration equal to the schedule length minus the transmission opportunity. Therefore, the step of the transmission opportunity plus the step of the deterministic backoff has a deterministic duration which is equal to the schedule length. When there are no transmission errors, no collisions and no missed acks, the duration of the whole cycle is deterministic. Only when there are collisions or errors, the additional step that extends the backoff for a random amount of time is included.

The random behaviour introduces a change in the schedule that may resolve the conflicts. The underlying philosophy is that when the schedule is not collision-free, there is at least one station that behaves randomly to force a schedule change. When a collision-free schedule is reached, the behaviour of all the nodes becomes deterministic to stay in that collision-free schedule.

The operation of the protocol is exemplified in Fig. 6. STA1 and STA2 are contending for the medium to transmit packets to each other and collide in their first transmission attempt.



**Fig. 6** An example of the operation of the protocol with two contending stations.

They are using a schedule length equal to four time units, and the transmission opportunity duration is one time unit. After they complete the transmission opportunity, they back off for a deterministic time equal to three units of time (the schedule length minus the transmission opportunity duration). Neither one of the two stations has received a positive ACK by the time they finish their deterministic backoff. Consequently, they back off for an additional random time that is exponentially distributed.

STA1 succeeds in its next transmission attempt. After the transmission opportunity, STA1 deterministically backs off for three units of time. When the deterministic backoff is completed, STA1 has not received any ACK packet, and therefore it has to extend its backoff period for a random amount of time. During the random backoff of STA1, STA2 finishes its own random backoff and successfully transmits. This last transmission includes an ACK for STA1, which is denoted in the figure by a small circle at the beginning of the transmission. Then STA1 successfully transmits, including an ACK for STA2. From this point on, the behaviour of the system is completely deterministic and periodic. The period is equal to the schedule length, and the systems operates in a collision-free fashion.

In our example, when the collision-free operation is reached, the ACK for a given packet arrives always before the deterministic backoff of the packet's transmitter expires. Therefore, the packet's transmitter is allowed a new transmission opportunity as soon as its deterministic backoff expires, and the system keeps working in a deterministic fashion.

#### 4.3 Stickiness

The example provided in Fig. 6 illustrates a simple case in which the transmitter and the receiver share the same schedule length. In complex topologies, it can also be the case that different stations use different schedule lengths. If that is the case, a station may miss an ACK after the deterministic backoff has elapsed because of

the fact that the receiver is using a longer schedule length. A missed ACK will be followed by a random exponential backoff that moves the station back to its random behaviour. To prevent this situation, the protocol needs to be modified to accept that some of the ACKs may be delayed. This can be accomplished by using a sticky variant of the protocol. Stickiness has already been used in [14] and [5] in the context of slotted networks, with the aim of reducing the duration of the transient state. A sticky protocol with a degree of stickiness equal to two will continue to operate deterministically even if one out of every two ACKs is missed. The flowchart of the sticky protocol (with a stickiness degree equal to two) is depicted in Fig. 5(b). Depending on the topology of the network, a larger degree of stickiness might be needed.

If stickiness is applied, it would be applied to all the transmissions, regardless of the destination node. The use of stickiness has some implications on the overall behaviour of the MAC protocol. As an example, stations would have multiple transmitted and not acknowledged packets. The acknowledgement timeouts should be adapted accordingly and the additional delays would affect the time required to converge to collision-free operation. For example, if a stickiness degree equal to two is used, the time-out has to be set to two times the schedule length. This change will necessarily increase the reaction time to detect and correct schedule configurations in which collisions occur.

#### 4.4 Convergence to collision-free operation

If the interference graph of the network topology is strongly connected, it is guaranteed that the collision-free schedule will be reached in finite time [7]. In a schedule that is not collision-free, there is at least one station that is behaving randomly because it does not receive ACKs for its packets. This station will change the position of its transmission opportunity within the schedule until a collision-free schedule is reached.



If that is not possible, the station will keep using a random backoff and, sooner or later, as a result of either delay or interference, some other station will miss one or more ACKs and move back to the random mode of operation and use random backoff. Since this can result in a cascading effect, a station that behaves randomly can trigger a schedule change that affects all the stations in the network. Therefore, the only stable state is the collision-free operation.

The interference graph is always strongly connected when data flows are bidirectional, otherwise additional measures should be taken. A possibility is to require stations to keep track of the data flows addressed to them, and use a random backoff if they do not receive such flows for the duration of several schedule lengths. The goal is to ensure that from any state of the network, it is possible to find a sequence of transmissions and collisions that drive the network to a collision-free operation.

For a given topology, to show that the system will eventually converge to collision-free operation we require that, for any possible state of the system  $c$ , there exists a chain of events of finite duration  $t_c$  which can bring the system to collision-free operation with non-zero probability  $p_c$ . Let  $t_c^*$  denote the largest  $t_c$  and  $p_c^*$  the minimum  $p_c$  over all possible states  $c$ . Then, for any given initial state of our network, the probability of convergence is at least  $p_c^*$  after a time interval  $t_c^*$ . After  $N$  time intervals, the probability of the network not having converged is not larger than  $(1 - p_c^*)^N$  and, as  $N$  goes to infinity, the probability that the network has not converged goes to zero. Consequently, the network converges with probability one.

The proof for the general problem of decentralized constrained satisfaction that directly applies to slotted scenarios is presented in [13].

#### 4.5 Stations transmitting multiple flows

So far we have assumed that each station is transmitting data to another station. It is common in multi-hop packet radio networks that a station has data to transmit to several neighbours. In that case, an instance of the backoff protocol should be executed for each of the destinations. This is similar to the different backoff instances used by different traffic classes in EDCA. As in EDCA, having multiple backoff instances may result in internal collisions that can be easily resolved.

We have already considered the possibility that a station has different outgoing flows in the computation of the schedule length. The parameter  $O_i$  in (2) is precisely the number of outgoing flows of station  $i$ .

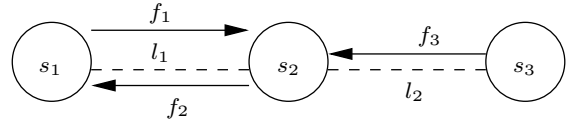


Fig. 7 Topology of a simple packet radio network.

Note that the protocol proposed in the present paper is different from the one in the original workshop paper [7]. The motivation of the changes is to take into account the existence of ACKs. The results presented in the next section show that the protocol suggested here converges faster to a collision free operation than the original one. The behaviour of the present protocol in the steady state is exactly the same as the one in [7] and for this reason we can reuse the steady state performance metrics of that paper.

### 5 Performance and simulation results

Since the transmission opportunity is normalized to one, the fraction of channel time used for transmission by a station  $i$  in the steady state is

$$\theta_i = \frac{O_i}{T_{\sigma_i}}, \quad (4)$$

where  $O_i$  is the number of outgoing flows in station  $i$  and  $T_{\sigma_i}$  is the schedule length of that station.

For illustration purposes, we will use the topology in Fig. 7 with three stations, two links and three data flows. Each station sees a total of three flows in its neighborhood. Using (1) and (2) we obtain that the schedule length for all the stations in this particular example is  $4(1 + \epsilon)$ . Then we can use (4) to compute the fraction of channel time devoted to successful transmissions by each of the stations and other network performance metrics (fairness, aggregate throughput, and proportional fairness).

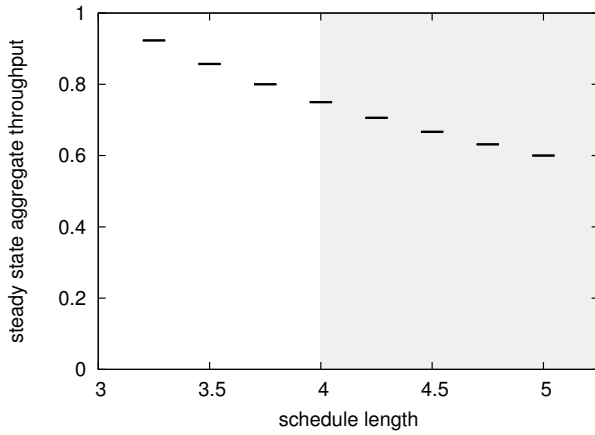
In Table 2 we compare the performance of the protocol described in the previous section, that we call self-configuring learning Aloha (scl-Aloha) to the Aloha protocol configured with optimal contention parameters to maximize proportional fairness [20].

Table 2 shows the values of  $\theta_1$ ,  $\theta_2$  and  $\theta_3$  for both Aloha and scl-Aloha. The details on the computation of the values for Aloha can be found in the appendix in [7]. The last three columns represent three different performance metrics: Jain's fairness index [19]

$$JF = \frac{(\theta_1 + \theta_2 + \theta_3)^2}{3(\theta_1^2 + \theta_2^2 + \theta_3^2)}, \quad (5)$$

aggregate throughput

$$AT = \theta_1 + \theta_2 + \theta_3, \quad (6)$$



**Fig. 8** Steady state aggregate traffic of the example network for different values of the schedule length.

and proportional fairness [21]

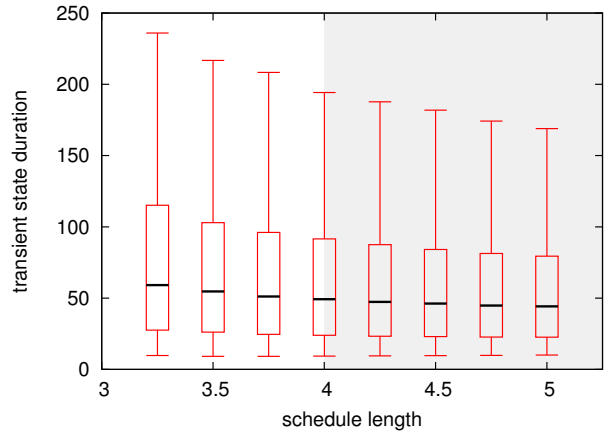
$$PF = \log \theta_1 + \log \theta_2 + \log \theta_3. \quad (7)$$

It can be observed that, in the steady state, scl-Aloha outperforms Aloha if  $\epsilon$  in (1) is appropriately chosen. We compare our protocol to optimally configured Aloha and not to CSMA, because CSMA is highly vulnerable to the hidden terminal problem that appears in multi-hop packet radio networks. In our particular example, STA1 and STA3 cannot carrier-sense each other and they would continuously collide.

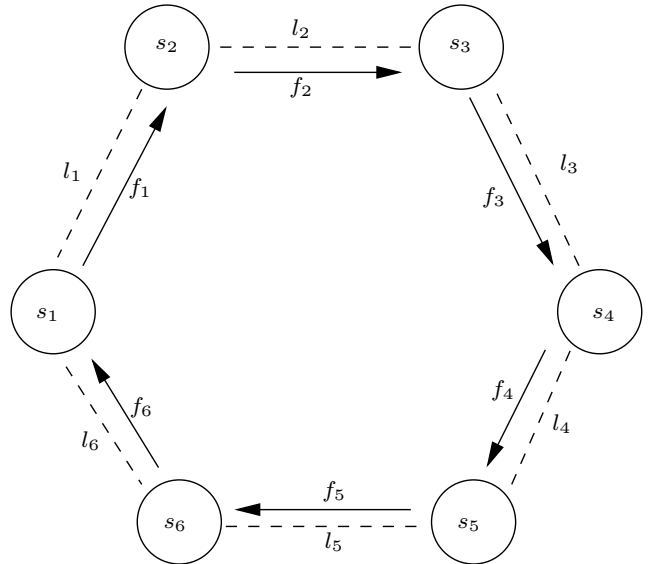
Fig. 8 depicts the aggregate throughput of our example network for different schedule lengths ranging from 3.25 to 7 in increments of 0.25. Throughput is measured in terms of successful transmissions per unit of time. Although any schedule length which is strictly larger than three would be in principle feasible, the schedule length selection process described in the previous section restricts the possible values of the schedule length to those that are larger than four (four is the smaller power of two that is strictly larger than three). The values larger than four are represented as a shaded area in the plot. This shaded area covers all schedule lengths that are equal to  $4(1 + \epsilon)$  for some positive value of epsilon. The smaller the value of  $\epsilon$ , the larger the aggregate throughput.

In order to assess the duration of the transient state, we have used a simple simulator that builds upon the SimPy [25] simulation framework for python<sup>1</sup>. We sweep schedule lengths from 3.25 to 5.00 in steps of 0.25. In each step, we run one thousand simulations

<sup>1</sup> This simulator includes all the assumptions and idealities discussed in the previous sections and it is available for download at [www.jaumebarcelo.info/barcelo2011emp/simulator.py](http://www.jaumebarcelo.info/barcelo2011emp/simulator.py). Contact the first author for more details and the scripts required to generate the plot.



**Fig. 9** A box-and-whiskers plot of the distribution of the absorption time for different schedule lengths. The 5, 25, 50, 75 and 95 percentiles are represented.



**Fig. 10** A six node packet radio network.

and compute the 5, 25, 50, 75, and 95 percentiles of the distribution of the duration of the transient state. The results are plotted in Fig. 9. Again, the schedule length selection protocol restricts the possible values of the schedule to those that are larger than four, which correspond to the shaded area in the plot.

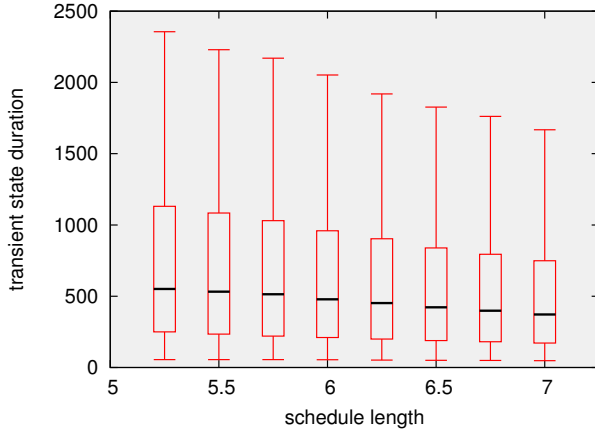
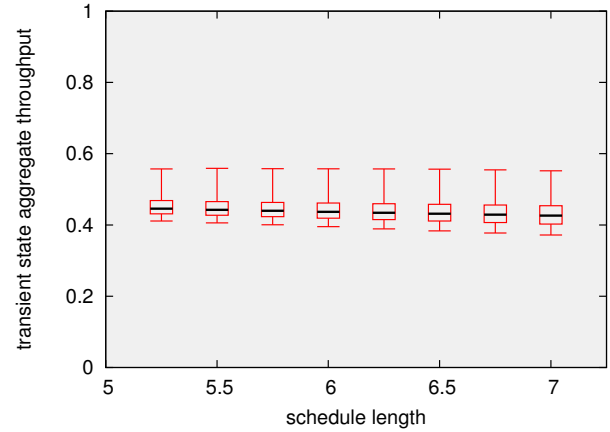
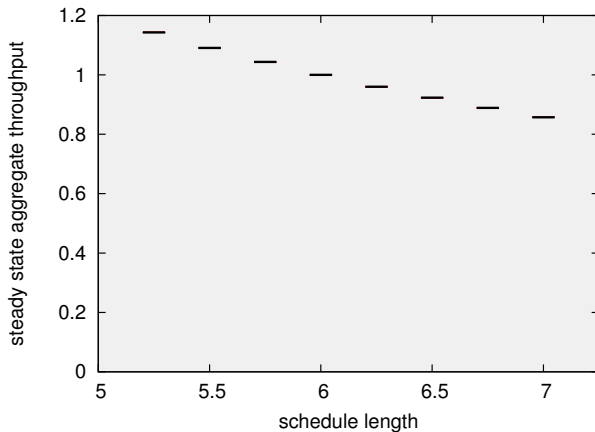
### 5.1 A six node ring topology

Fig. 10 represents a topology consisting of six nodes in a ring configuration. In this configuration there is also a hidden node problem that cannot be prevented using CSMA.

We test this topology using simulation and derive statistics on the time required for the network to con-

**Table 2** Performance Comparison

Contention	Per station throughput			Network performance		
	$\theta_1$	$\theta_2$	$\theta_3$	JF	AT	PF
Aloha	0.056	0.120	0.108	0.921	0.283	-7.234
scl-Aloha	$\frac{0.056}{4(1+\epsilon)}$	$\frac{0.120}{4(1+\epsilon)}$	$\frac{0.108}{4(1+\epsilon)}$	1	$\frac{0.283}{4(1+\epsilon)}$	$-4.159 - 3\log(1+\epsilon)$

**Fig. 11** A box-and-whiskers plot of the distribution of the absorption time for different schedule lengths. The 5, 25, 50, 75 and 95 percentiles are represented.**Fig. 13** Transient state aggregate traffic of the six node network for different values of the schedule length.**Fig. 12** Steady state aggregate traffic of the six node network for different values of the schedule length. Note that in this particular topology, it is possible that two stations simultaneously successfully transmit. For this reason, values of throughput higher than one are possible.

verge that are presented in Fig. 11. We consider schedule lengths ranging from 5.25 to 7 in increments of 0.25.

If optimal synchronization and scheduling were possible, two of the six stations could be transmitting at any time without interfering each other, which means an aggregate throughput equal to two. The throughput in the steady state that is obtained by the proposed algorithm using the mentioned schedule lengths is roughly one (See Fig. 12). Higher throughputs could be obtained at the expense of longer convergence times.

The performance that the network achieves during the transient state is substantially lower than the performance that is obtained in the steady state. Fig. 13 represents the statistics of the aggregate throughput that is achieved in the six node topology during the transient state. Even though the presence of some collisions has a clear impact on performance, it does not present the starvation problems associated to hidden node scenarios.

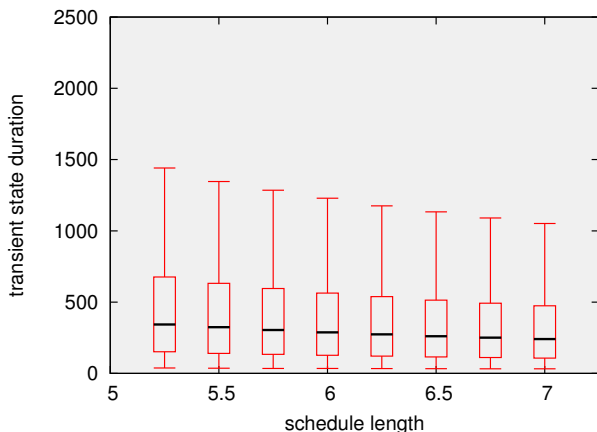
## 6 Refinements and future work

In this paper we have considered a very simple protocol that schedules the next transmission as a function of the result (either success or failure) of the last transmission. We have seen that, despite its simplicity, this protocol can reach collision-free operation and solve the MAC layer scheduling problem in multi-hop packet radio networks. A more sophisticated approach could converge more quickly to the collision-free schedule by including two additional rules in the protocol:

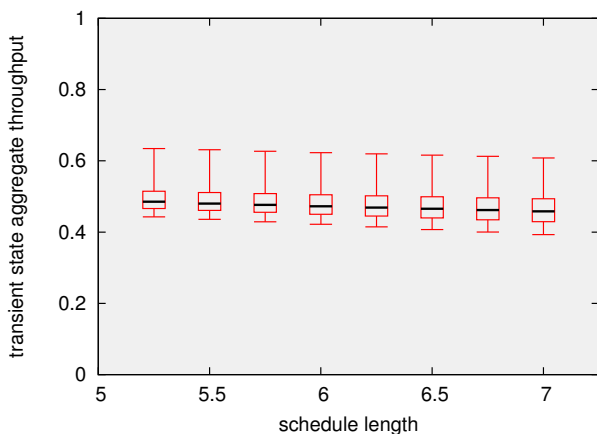
- Do not transmit while a packet is being received.
- Do not transmit to a station that it is engaged in a transmission.

Violating any of those two rules results in a certain collision.

From this perspective, the results presented in this paper should be considered a lower bound on the perfor-



**Fig. 14** Transient state duration statistics for the six node network and different values of the schedule length. The protocol combines CSMA and collision-free properties.



**Fig. 15** Transient state aggregate traffic of the six node network for different values of the schedule length. The protocol combines CSMA and collision-free properties.

mance that can be achieved by the family of collision-free MAC protocols.

As a first step towards collision-free protocols with better convergence properties, we have repeated the experiments with the six node ring topology using a hybrid protocol that combines both CSMA and the collision-free properties that have been described in the present paper. In particular, if a node senses the medium busy after finishing a backoff, it starts a new backoff of exponentially distributed length.

The results are presented in Figs. 14 and 15 and evidence a performance improvement, specially in terms of the time required to reach collision-free operation. The throughput in the steady state is not plotted, since it is exactly the same as in Fig. 12.

A restriction of the protocol as presented in this paper is the fact that the number of flows in one station's neighborhood is assumed to be known, and therefore

the nodes need to exchange *hello* messages when the network starts to operate or when the traffic patterns change (as described in [20]).

Information about the number of flows in the neighborhood can be estimated by overhearing the neighbours' transmissions and ACKs. The transmissions can be used to infer the number of outgoing flows while the ACKs make it possible to estimate the number of incoming flows.

In the presence of non-saturated flows, the stations with no packets to transmit might skip their turn while respecting the overall schedule. If some of the flows have special requirements, such as requiring twice as much bandwidth as the normal flows, they should be split in two different flows (with the same origin and destination station as the original flow) and execute two different instances of the backoff protocol to have access to two transmission opportunities in each schedule.

A topology change or a change in the traffic pattern may require a re-computation of the schedule and the schedule length. The applicability and the advantages of the protocol presented in this paper to dynamic networks is still a matter that requires further study.

Another aspect of interest is the combination of the proposed approach with existing mechanisms such as MCCA [17]. Our protocol could be used to find a collision-free schedule for MCCA beacons, which in turn could be used to make channel reservations for data transmissions.

## 7 Conclusion

In this paper we have described the possibility of distributively constructing a collision-free schedule in multi-hop packet radio networks. We have first discussed the problem of channel time fragmentation and explained that this problem can be alleviated by using delayed ACKs at the link layer. Then we have dealt with the distributed computation of the schedule length, requiring only the exchange of local information among neighbours. After the schedule length has been computed, the stations engage in the actual contention, using a deterministic backoff if a packet has been successfully transmitted (and acknowledged), and a random backoff otherwise. When the collision-free schedule is reached, all the stations behave deterministically and no channel time is wasted in the form of collisions. Therefore, the proposed protocol offers the possibility of making a more efficient use of the channel time. The collision-free operation is reached only after a transient state, the duration of which we assessed for two particular topologies using simulation.

**Acknowledgements** The authors are thankful to the anonymous reviewers for their constructive comments that helped to substantially improve the final version of this manuscript. This work has been partially funded by the Spanish Government (grant TEC2008-0655) and the European Commission (grant CIP-ICT PSP-2011-5). The views expressed in this article are solely those of the authors and do not represent the views of the Spanish Government or the European Commission.

## References

- Andreev, K., Boyko, P.: Simulation Study of VoIP Performance in IEEE 802.11 Wireless Mesh Networks. *MA-COM* pp. 139–150 (2010)
- Arikan, E.: Some Complexity Results about Packet Radio Networks. *IEEE Transactions on Information Theory* **30**(4), 681–685 (1984)
- Barcelo, J., Bellalta, B., Cano, C., Oliver, M.: Learning-BEB: Avoiding Collisions in WLAN. In: *Eunice* (2008)
- Barcelo, J., Bellalta, B., Cano, C., Sfairopoulou, A., Oliver, M.: Carrier Sense Multiple Access with Enhanced Collision Avoidance: a Performance Analysis. In: *ACM IWCMC* (2009)
- Barcelo, J., Bellalta, B., Cano, C., Sfairopoulou, A., Oliver, M.: Towards a Collision-Free WLAN: Dynamic Parameter Adjustment in CSMA/E2CA. In: *Eurasip Journal on Wireless Communications and Networking* (2011)
- Barcelo, J., Bellalta, B., Cano, C., Sfairopoulou, A., Oliver, M., Zuidweg, J.: Traffic Prioritization for Carrier Sense Multiple Access with Enhanced Collision Avoidance. In: *MACOM* (2009)
- Barcelo, J., Bellalta, B., Oliver, M., Banchs, A.: Collision Free Operation in Ad-Hoc Networks. In: *MACOM* (2011)
- Barcelo, J., Bellalta, B., Sfairopoulou, A., Cano, C., Oliver, M.: CSMA with Enhanced Collision Avoidance: a Performance Assessment. In: *IEEE VTC Spring* (2009)
- Barcelo, J., Lopez-Toledo, A., Cano, C., Oliver, M.: Fairness and Convergence of CSMA with Enhanced Collision Avoidance. In: *ICC* (2010)
- Chambers, B.: The grid roofnet: a rooftop ad hoc wireless network. Ph.D. thesis, Massachusetts Institute of Technology (2002)
- Choi, J., Yoo, J., Choi, S., Kim, C.: EBA: An Enhancement of the IEEE 802.11 DCF via Distributed Reservation. *IEEE Transactions on Mobile Computing* **4**(4), 378–390 (2005)
- Cicconetti, C., Lenzini, L., Mingozzi, E.: Scheduling and dynamic relocation for IEEE 802.11 s mesh deterministic access. In: *IEEE SECON*, pp. 19–27 (2008)
- Duffy, K.R., Bordenave, C., Leith, D.J.: Decentralized constraint satisfaction. *CoRR abs/1103.3240* (2011)
- Fang, M., Malone, D., Duffy, K., Leith, D.: Decentralised Learning MACs for Collision-free Access in WLANs. *Arxiv preprint arXiv:1009.4386v2* (2011)
- Gurewitz, O., Mancuso, V., Shi, J., Knightly, E.: Measurement and Modeling of the Origins of Starvation of Congestion-Controlled Flows in Wireless Mesh Networks. *IEEE/ACM Transactions on Networking* **17**(6), 1832–1845 (2009)
- He, Y., Yuan, R., Sun, J., Gong, W.: Semi-Random Back-off: Towards Resource Reservation for Channel Access in Wireless LANs. In: *IEEE ICNP*, pp. 21–30 (2009)
- Hiertz, G., Denteneer, D., Max, S., Taori, R., Cardona, J., Berlemann, L., Walke, B.: IEEE 802.11 s: The wlan mesh standard. *IEEE Wireless Communications Magazine* **17**(1), 104–111 (2010)
- Hui, K., Li, T., Guo, D., Berry, R.: Exploiting peer-to-peer state exchange for distributed medium access control. In: *IEEE ISIT*, pp. 2368–2372. *IEEE* (2011)
- Jain, R.: *The Art of Computer Systems Performance Analysis*. John Wiley & Sons New York (1991)
- Kar, K., Sarkar, S., Tassiulas, L.: Achieving Proportional Fairness Using Local Information in Aloha Networks. *IEEE Transactions on Automatic Control* **49**(10), 1858–1863 (2004)
- Kelly, F.: Charging and rate control for elastic traffic. *European transactions on Telecommunications* **8**(1), 33–37 (1997)
- Khan, Z., Lehtomaki, J., Mustonen, M., Matinmikko, M.: Sensing order dispersion for autonomous cognitive radios. In: *CROWNCOM*, pp. 191–195. *IEEE* (2011)
- Krasilov, A., Lyakhov, A., Safonov, A.: Interference, Even with MCCA Channel Access Method in IEEE 802.11s Mesh Networks. In: *MASS*, pp. 752–757 (2011)
- Martorell, G., Riera, F., Femenias, G., Barcelo, J., Bellalta, B.: On the performance evaluation of CSMA/E2CA protocol with open loop ARF-based adaptive modulation and coding. In: *European Wireless* (2012)
- Mueller, K.: *SimPy Documentation*. <http://simpy.sourceforge.net/> (2004). [Online; accessed 05-December-2011]
- Oliver, M., Zuidweg, J., Batikas, M.: Wireless Commons Against the Digital Divide. In: *IEEE ISTAS* (2010)
- Ramanathan, S., Lloyd, E.: Scheduling Algorithms for Multihop Radio Networks. *IEEE/ACM Transactions on Networking* **1**(2), 166–177 (1993)
- Sommer, P., Wattenhofer, R.: Gradient Clock Synchronization in Wireless Sensor Networks. In: *ACM/IEEE IPSN* (2009)
- Tian, X., Chen, X., Ideguchi, T., Okuda, T.: Improving protocol capacity by scheduling random access on WLANs. *Telecommunication Systems* **37**, 19–28 (2008)
- Tobagi, F.: Modeling and Performance Analysis of Multihop Packet Radio Networks. *Proceedings of the IEEE* **75**(1), 135–155 (1987)
- Yi, Y., De Veciana, G., Shakkottai, S.: MAC Scheduling With Low Overheads by Learning Neighborhood Contention Patterns. *IEEE/ACM Transactions on Networking* (2010)